

Apellidos:

Nombre:

Matrícula:

Examen

105000016 - Programación para Sistemas Grado en Ingeniería Informática (2009)

Lenguajes y Sistemas Informáticos e Ingeniería de Software

Facultad de Informática

Universidad Politécnica de Madrid

Curso 2014/2015 - Junio 2015

Normas

- El examen puntúa sobre **12 puntos**.
- La duración total del mismo es de **55 minutos**.
- Se deberá tener el DNI o el carnet de la UPM en lugar visible.
- No olvidar rellenar **apellidos, nombre y número de matrícula** en cada hoja.
- La solución al examen se proporcionará antes de la revisión.

Cuestionario

- (1 punto) 1. Al final de un script se desea devolver un número (exit status) indicando cómo ha acabado dicho script. Indique el comando a usar en la forma adecuada para devolver un exit status 1.

Solución:

exit 1

- (1 punto) 2. Indique qué denota \$@ en un script. (Describalo obligatoriamente en una línea como máximo.)

Solución: Lista de argumentos. Copia exacta de la lista de argumentos.

- (1 punto) 3. Suponiendo que el fichero ./fich.txt no existe, indique la salida estándar que resultará al ejecutar en Bash lo siguiente:

```
test -e ./fich.txt && echo 'Hola'
```

Solución: Salida vacía.

- (1 punto) 4. En Unix, Indique cómo conectar comando1 y comando2 con un 'pipe', de forma que la salida estándar de comando1 sea la entrada estándar de comando2.

Solución:

```
comando1 | comando2
```

- (1 punto) 5. Se pide escribir el comando necesario para compilar el fichero de código (que se ha escrito en el fichero principal.c), si dicho fichero utiliza la librería stdio.h y la librería auxiliar.h vista en clase (implementada en auxiliar.c). El comando pedido debe generar un fichero ejecutable llamado: ejecutable.out

Solución: gcc principal.c auxiliar.c -o ejecutable.out

- (1 punto) 6. Dado el siguiente código:

```
#include <stdio.h>
main() {
    int vector [10];
    int *ptr = &vector[2];
    int cont=0;
    *ptr = 10;
    for(cont=1; cont<10; cont++) {
        vector [cont] = cont*10;
    }
    for(cont=0; cont<6; cont=cont+2) {
        printf("%d-", *(ptr+cont));
    }
}
```

Indicar cual será la salida por pantalla al ejecutar dicho código.

Solución: 20-40-60-

- (1 punto) 7. Dado el siguiente código C

```
#include <stdio.h>
int valor1 = 1;
void cambio(int *valor1, int *valor2);
```

```
main() {
    int valor2 = 2;
    printf("%d_", valor1);
    cambio(&valor1, &valor2);
    printf("%d_", valor2);
    cambio(&valor1, &valor2);
    printf("%d_", valor1);
}
void cambio(int *num1, int *num2) {
    int valor1 = 3;
    *num2 = *num1;
    *num1 = 4;
}
```

Indicar cual sera la salida por pantalla al ejecutar dicho codigo.

Solución: 1 1 4

(1 punto) 8. Dado el siguiente fragmento de codigo:

```
#include <stdio.h>
#include <stdlib.h>
main(int argc, char *argv[]){
    FILE *fp;
    if (argc<2){
        printf ("Uso:_programa_nombre_fichero\n");
        exit (1);
    }
    fp = fopen (...(1)...);
    if (...(2)...) {
        printf ("Fichero_no_existe");
        exit (1);
    }
    char linea[2048];
    ...(3)...
    printf ("%s", linea);
    fclose ( fp );
    return 0;
}
```

Escribir las 3 partes de codigo faltante para que el programa lea de un fichero, pasado como primer argumento al programa, la primera linea que haya en el fichero y la escriba por pantalla.

Solución: ...(1)... argv[1], r"
...(2)... fp==NULL
...(3)... fgets (linea , 2048 , fp);

- (1 punto) 9. Suponiendo la siguiente definicion de estructura para numeros complejos,

```
typedef struct {  
    double real;  
    double imaginaria;  
} T_COMPLEJO
```

escribir la sentencia necesaria para crear un vector que permita almacenar hasta 10 nsmeros complejos.

Solución:

```
T_COMPLEJO vector[10];
```

- (1 punto) 10. Escribir las sentencias necesarias para reservar memoria dinamica suficiente para almacenar una cadena de 30 caracteres. Incluyendo la comprobacion de que la solicitud de memoria dinamica se ha procesado con exito.

Solución:

```
char * cadena;  
cadena = (char *) malloc (31);  
if (cadena == NULL) {  
    exit (-1) ;  
}
```

- (1 punto) 11. Dado el siguiente codigo C, indica la respuesta correcta:

```
#include <stdlib.h>  
#include <stdio.h>  
struct punto  
{  
    int x;  
    int y;  
};  
struct punto *f()  
{  
    struct punto *p;  
p=(struct punto *)malloc(sizeof(struct punto));  
    (*p).x=5;  
    (*p).y=(*p).x;  
    return p;  
}  
void main()  
{  
    struct punto *p1;  
p1=f();
```

Apellidos:

Nombre:

Matrícula:

```
printf("%d, %d", p1->x, p1->y);  
}
```

- a) Escribe: 5, 0
- b) Escribe: 5, 5
- c) Se produce un error en la ejecución, segmentation fault
- d) Escribe: 0, 0

Solución: b)

- (1 punto) 12. Se desea depurar un programa en C, cuyo código se encuentra en el fichero ejemplo.c. Escribir las acciones a realizar para compilar dicho programa, arrancar el depurador, establecer un punto de ruptura en la línea 20, ejecutar el programa, que prosiga la ejecución después de pararse en el punto de ruptura establecido y finalmente salirse del depurador.

Solución: gcc -g -o ejemplo ejemplo.c
gdb ejemplo
breakpoint 20
run
continue
quit